CLAIMS

What is claimed is:

1. A computing system employing VIA for data communications from a VIA enabled software application on said computing system with data destinations other than said VIA enabled software application wherein a path for said data communications is through at least one Network Interface Card (NIC) with at least one port connection per each NIC, said computing system comprising:

a management component for tracking a plurality of NIC connections resident on said computing system said management component having a name for each NIC it is tracking but showing only a single NIC name to said VIA enabled software application.

2. The computer system of claim 1 wherein said management component comprises:

a Physical NIC data structure for keeping data needed and used to manage each instance of a NIC,

a Logical NIC data area useable by a Kernel Agent on said host computer system to manage a plurality of logical NIC objects made visible to the Virtual Interface Provider Library and to the Application program, including a link to all Physical NICs for devices within said Logical NIC, said Logical NIC having a Master Completion Queue for each Physical NIC which reference any completion queue associated with said Logical NIC,

a protection tag table data structure for storing locally unique protection tag for each application , providing that each protection tag is a unique value within a NIC so that only one application's privileges are related to any one of said protection tags.

3. The computer system of claim 2 wherein the protection tag table, and a memory registration table are data structure components of said Logical NIC data area.

4. The computer system of claim 3 wherein a completion queue notification request area (CQNRA) is set up for each Logical NIC, so that the CQNRA can be responsive to completions of any master completion queue associated with any Physical NIC.

5. The computer system of claim 4 wherein a failure of any NIC device within a logical NIC is not visible to an application program using said Logical NIC so long as an additional NIC device is operational.

6. A data management component for tracking a plurality of NIC connections resident on a computing system, wherein said computing system employs VIA for data communications from a VIA enabled software application on said computing system with data destinations other than said VIA enabled software application wherein a path for said data communications is through at least one Network Interface Card (NIC) with at least one port connection per each NIC, said data management component comprising:

a Physical NIC data structure for keeping data needed and used to manage each instance of a NIC,

a Logical NIC data area useable by a Kernel Agent on said computer system to manage a plurality of logical NIC objects made visible to the Virtual Interface Provider Library and to the Application program, including a link to all Physical NICs for devices within said Logical NIC, said Logical NIC having a Master Completion Queue through which all completions for all Physical NICS associated with said Logical NIC,

a protection tag table data structure for storing locally unique protection tag for each application , providing that each protection tag is a unique value within a NIC so that only one application's privileges are related to any one of said protection tags.

7. The data management system of claim 6 wherein the protection tag table, and a memory registration table are data structure components of said Logical NIC data area.

8. The data management system of claim 7 wherein a completion queue notification request area (CQNRA) is set up for each Logical NIC, so that the CQNRA can be responsive to completions of any master completion queue associated with any Physical NIC.

9. The data management system of claim 8 wherein a failure of any NIC device within a logical NIC of said computer system is not visible to an application program on said computer system with said data management system is using said Logical NIC so long as an additional NIC device is operational.

10. A method for providing physically independent network interface cards for a computer system having a plurality of network interface cards (NICs) so as eliminate the need of an application program on said computer system to identify a particular one of said plurality of NICs, said method comprising;

establishing a process for NIC Attribute Handling which provides a representation of a physical NIC to said application program and a data structure for use by said process to track NIC-application program interaction, Network Address Visibility, Load balancing, VI Creation, Memory Registration and Memory Deregistration, Changing Memory Attributes, Dialog Establishment, Descriptor Posting, Descriptor Processing, Work Queue Notification, and Completion Queue Notification, and

establishing a process for returning an alias address value responsive to a VipQueryNic from said application program.

11. The method of claim 10 wherein the alias address value is uniquely qualified for each NIC.

12. The method of claim 10 wherein said alias value may be user-configured.

13. The method of claim 10 further comprising:
determining which NIC to assign to an application program requiring the use of one to establish a Virtual Interface, at the time said requirement is requested, said assignment being determined based upon a Load balancing algorithm.

14. The method of claim 13 wherein said Load Balancing algorithm is may be a simple round-robin VI assignment among NICs having network connectivity to the remote destination.

15. The method of claim 13 wherein said Load Balancing algorithm establishes a bias to select NICs having lighter loads.

16. The method of claim 15 wherein said bias can be determined in real time and data transfer work shunted to additional NICs where a selected NIC is a bottle-neck.

17. The method of claim 10 wherein a completion queue notification request area (CQNRA) is set up for Logical NIC, so that the CQNRA responds to completions of any master completion queue associated with any Physical NIC, allowing a Kernel Agent to notify the application program of a completion by a Physical NIC related to a virtual interface.